

Průzkumné vozidlo

Konečný Jaromír · Elektrotechnika, Študentské práce

11.10.2010



Práce se zabývá návrhem a sestavením platformy mobilního robotického zařízení sloužícího jako průzkumné vozidlo. Vozidlu je možné z ovládacího panelu zadat místo, kam má dojet, průzkumné vozidlo na toto místo dojede bez zásahu člověka. V terénu se orientuje podle vektorových map a senzorického systému. Zároveň se vozidlo umí vyhýbat překážkám. Motivací práce je postupné vytvoření platformy, která bude schopna prozkoumat neznámé prostředí, vyhledat zraněné a navázat s nimi kontakt.

1. Průzkumné vozidlo

Průzkumné vozidlo lze použít pro orientaci jak v otevřeném, tak uzavřeném prostoru. Je vybaveno skupinou senzorů pro měření vzdálenosti, teploty, tlaku a jiných, které umožňují lepší detekci prostředí a které pomáhají odhadnout případné nebezpečí. Zařízení je schopno se přesunout na požadovanou pozici.

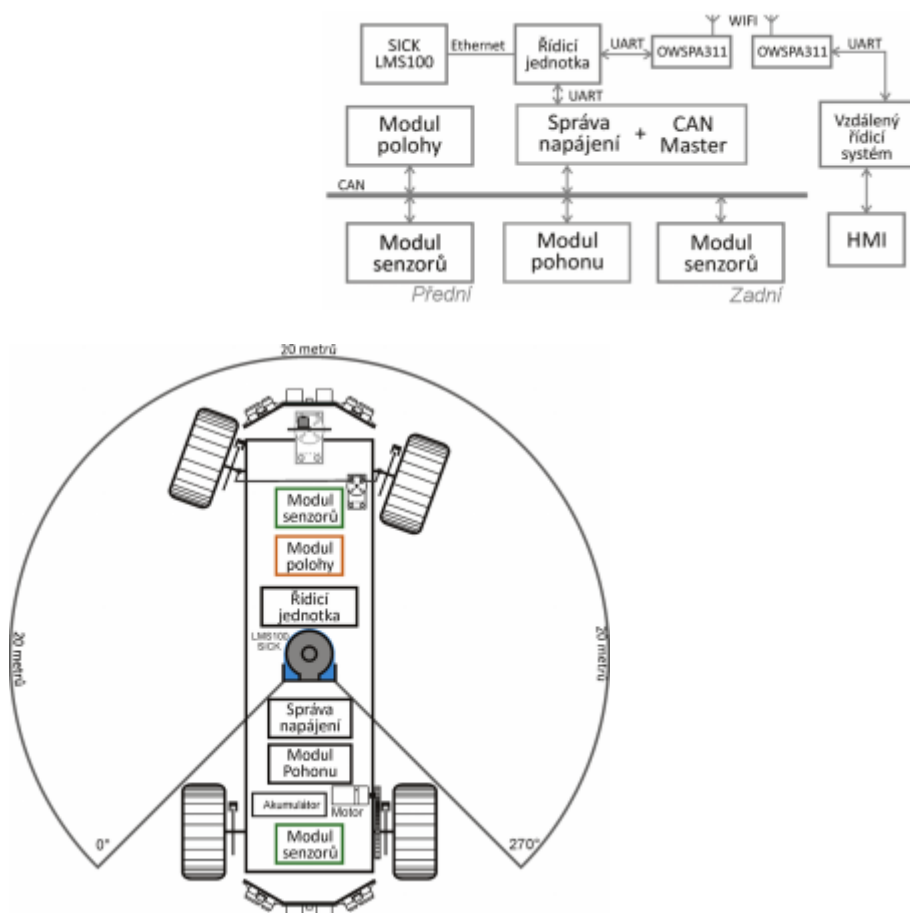
Robotický model je sestaven z podvozku RC automobilu Baja 5B SS. Jedná se o sportovní model automobilu s velikostí 1:5. Je přibližně 85 cm dlouhý a 60 cm široký. Podvozek byl osazen benzínovým motorem, který byl nahrazen elektromotorem. Díky tomuto klesla váha podvozku a do uvolněného místa je zabudována elektronika. Model disponuje diferenciálem, převodovkou o celkovém poměru cca. 1:16, tlumiči a ovládním řízením předních kol pomocí servo-motoru. Model je dostatečně velký a robustní a tím umožňuje montáž všech potřebných částí, jako jsou baterie, motor, elektronika a snímací senzory.

1.1 Architektura

Řízení průzkumného vozidla je koncipováno jako distribuovaný řídicí systém. Ten je složen z několika samostatných modulů, které jsou umístěny přímo na vozidle a většina z nich spolu komunikují prostřednictvím průmyslové sběrnice CAN a CANOpen aplikační vrstvě. Řídicí jednotka, tedy hlavní modul vozidla, není připojena ke sběrnici CAN, ale přes sériové rozhraní k podřízeným modulům. Jiným sériovým rozhraním je připojena k WiFi modulu. Vozidlo je složeno z několika modulu:

- Hlavní řídicí jednotka
- Jednotka správy napájení
- Modul pohonu
- Přední a zadní modul senzorů

- Modul polohy



Obr. 1: Blokové schéma architektury průzkumného vozidla a rozmístění komponent

Každý podrizený modul je umístěn co nejbližší CAN sběrnice, aby se snížila délka vodičů a EMC šum. Rozmístění modulů po vozidle je zobrazeno na obrázku výše. Na blokovém schématu je vidět i část se vzdáleným řídicím systémem, který komunikuje s řídicí jednotkou vozidla přes bezdrátovou síť WiFi vytvořenou moduly OWSPA311. Moduly se chovají jako bezdrátová sériová linka. Průzkumné vozidlo je vybaveno sadou senzorů:

- GPS modul s SiRF Starr III architekturou
- magnetický senzor, který slouží jako kompas
- laserový senzor SICK (dosah 20 metrů - 270° v kroku 0.25°)
- infra kamera pro měření infra teploty
- dva ultrazvukové senzory pro měření vzdálenosti (dosah až 3 m)
- 4 optické senzory pro měření vzdálenosti na rozích vozidla (dosah až 0.8 metrů)
- tlakový senzor pro získání nadmořské výšky
- teplotní senzor
- čidlo plynu a 2 čidla osvětlení
- 3 osý akcelerometr
- Inkrementální senzory pro měření otáček všech kol

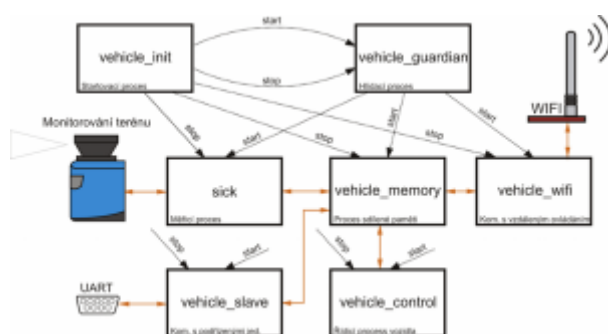
1.2 Hlavní řídicí jednotka

Hlavní řídicí jednotka je postavena na vývojové sadě i.MX31 LiteKit, která je založena na ARM architektuře. Jádrem desky je výkonný procesor Freescale

i.MX31. Vývojová deska je vybavena sadou rozhraní (Ethernet, sériový port, USB, SPI, SD, CF, IDE, atd). Pro podporu ladění je zařízení vybaveno sériovým a Ethernetovým konektorem. Platforma i.MX31 byla oživena s operačním systémem Timesys Linux . Proces oživení v sobě zahrnuje úpravu zavaděče, zkompilování a vypálení jádra operačního systému do paměti, dále pak zkopírování souborového systému na SD paměťovou kartu. Řídící aplikace s ostatními procesy jsou rovněž uloženy na externím paměťovém médiu (SD).

Vývoj aplikace je prováděn ve vývojovém prostředí TimeStorm v programovacím jazyce C++ na hostitelském počítači s operačním systémem Fedora Linux . Prostředí na vývoj zahrnuje podporu balíčků (BPS) pro i.MX31 LiteKit. Z počátku vývoje se souborový systém nacházel pouze na hostitelském počítači (služba NFS) a později byl nakopírován přímo na SD kartu. Hlavní řídicí aplikace využívá několik spolupracujících procesů. Tato metoda dovoluje hlavnímu řídicímu systému dynamicky spustit, zastavit nebo nahradit část aplikace bez nutnosti zastavení aplikace. Použité procesy:

- vehicle_init - inicializace aplikace
- vehicle_guardian - hlídač spuštěných procesů (při pádu procesu, jej restartuje)
- vehicle_memory - proces poskytující sdílenou paměť
- vehicle_slave - komunikace s podřízenými jednotkami
- vehicle_wifi - komunikace se vzdáleným řídicím systémem
- sick - čtení dat z laserového senzoru, zapisuje hodnoty do sdílené paměti
- vehicle_control - hlavní řízení, výpočet vhodné cesty, mapy



Obr. 2: Blokové schéma běžících procesů

Po resetu i.MX31 LiteKit (zapnutí) se spustí Logic Loader, který načte operační systém, ten si připojí souborový systém z SD karty a poté spustí proces `vehicle_init`, který následně spustí ostatní procesy.

1.3 Podřízené jednotky

Podřízené řízení a měřicí desky jsou založeny na průmyslových mikrokontrolérech Freescale HCS12 s kooperativním operačním systémem FreeRTOS. Tyto desky jsou naprogramované v programovacím jazyce C pomocí prostředí CodeWarrior IDE. Každá podřízená jednotka se chová jako CANOpen uzel. Hlavní jednotka komunikace (Master) je modul správy napájení. Tento modul zapíná řídicí jednotku a systém senzorů v případě dobrého stavu baterie.

Jednotka správy napájení

Slouží pro distribuci elektrické energie z akumulátoru do jednotlivých modulů

na vozidle. Řídí nabíjení, dobíjení, sleduje stav baterie při vybíjení a dále jej vyhodnocuje. Je také datovým mostem mezi CAN sběrnici a nadřazeným systémem.

Modul polohy

Poskytuje řídicí jednotce informace o aktuální pozici vozidla. Tu vypočítává z globálních a relativních informací o poloze z připojených senzorů (GPS, magnetický kompas, odometrie). Mimo to získává informace okolním prostředím jako teplota, tlak, koncentrace výbušných látek.

Moduly senzorů

Na vozidle se nacházejí dva tyto moduly. Ty fungují jako sběrné uzly, které shromažďují informace ze svého okolí a ty poskytují je ostatním modulům. Jsou vybaveny snímači pro měření vzdálenosti (optické a ultrazvukové). Modul nacházející se na přední části vozidla zároveň ovládá natočení předních kol.

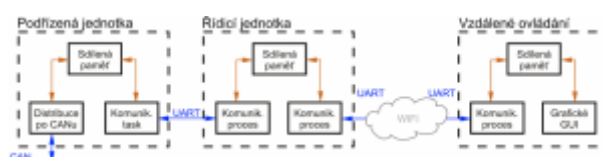
Modul pohonu

Obsahuje výkonovou část pro řízení pohonu. Implementuje algoritmus pro regulaci otáček motoru, proudové omezení atd. Požadovaná rychlost pojezdu je ovládána prostřednictvím příkazů z nadřazeného systému.

1.4 Princip komunikace

Komunikace s podřízenými jednotkami probíhá po sériovém rozhraní, protože i.MX31 nedisponuje CAN rozhraním. Celý koncept komunikace je řešen pomocí sdílené paměti, ve které se nachází data, která se mezi jednotkami a ovládáním vyměňují. Na straně vzdáleného ovládání je identický blok sdílené paměti jako v řídicí jednotce a v modulu správy napájení (battery management), který se stará o přeposílání dat na sběrnici CAN. V modulu správy napájení je datový blok zjednodušen. V řídicí jednotce jsou dva samostatné procesy. Jeden se stará o komunikaci s nadřazeným zařízením a data, která přijme a zapíše bez jakékoliv znalosti jejich obsahu do sdílené paměti a naopak jiná data stále cyklicky posílá vzdálenému modulu. Druhý proces funguje obdobně, ale takto komunikuje s modulem správy napájení.

Popišme nyní situaci např. ručního zapnutí světel. Na začátku jsou světla vypnuta. Uživatel vzdáleného zařízení stiskne tlačítko pro zapnutí světel a tento stisk zapíše do sdílené paměti umístěné ve vzdáleném zařízení, že světla mají být zapnuta. Proces, který se stará o komunikaci s řídicí jednotkou, cyklicky posílá tato data. Jakmile řídicí jednotka vozidla přijme nová data, uloží si je do své sdílené paměti. Po přijmutí dat však s nimi nic nedělá. Proces, který se stará o komunikaci s podřízenými moduly vysílá data ze sdílené paměti a vyšle i pokyn k zapnutí světel. Ten se dostane až do správy napájení, kde je vydán příkaz modulu řízení světel pro zapnutí světel a tím je úkol splněn. Tímto způsobem je možné pospat tok dat v celém zařízení.



Obr. 3: Schéma toku dat

2. Vzdálený řídicí systém

Vzdálené řízení je druhou částí mobilního zařízení. Poskytuje vzdálenou obsluhu přes Human-Machine Interface (HMI). Je to ruční zařízení střední velikosti s LCD dotykovým displejem a joystickem. Řídicí systém je ukázán na obrázku pod textem.



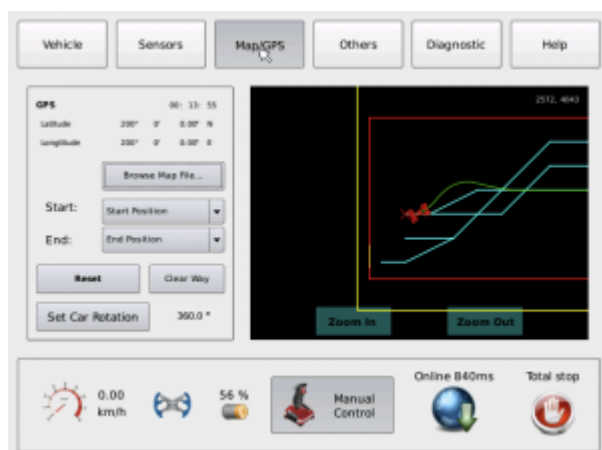
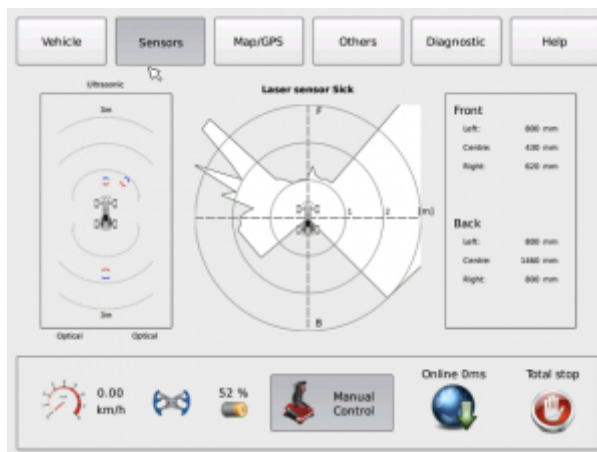
Obr. 4: Vzdálený řídicí systém

Vzdálený řídicí systém poskytuje:

- monitorování a řízení průzkumného vozidla
- zobrazování informací ze senzorů
- vložení a aktualizace mapy
- nastavení cílové pozice pro automatický režim
- manuální řízení prostřednictvím joysticku (gamepadu)
- diagnostiku všech údajů, grafy historických hodnot a nápovědu

Uživatel může mít buď plnou kontrolu nad řízením autonomního průzkumného vozidla, nebo může pouze vyslat cílovou pozici. Uživatel ovládá vozidlo přes barevný dotykový panel. Aplikace poskytuje intuitivní uživatelské rozhraní se sadou tlačítek, textových displejů a obrazovek. Mezi řídicí jednotkou vozidla a vzdáleným řídicím systémem se přenáší velké množství dat. Tato data byla rozdělena podle funkce a seříděna, aby například společné hodnoty senzorů byla s ostatními senzory. Existují zde ale i data, která by měl mít uživatel stále na očích a nemusel je složitě hledat.

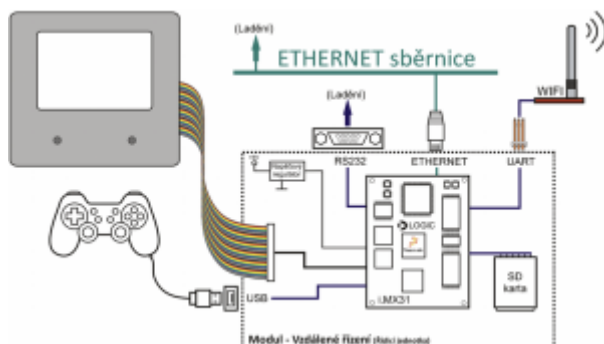
Pod textem jsou ukázány dvě obrazovky z aplikace s graficko-uživatelským rozhráním. První obrázek zobrazuje situaci senzorů pro měření vzdálenosti (laserový, optický a ultrazvukový). V této situaci se něco nachází velmi blízko před vozidlem. Volný prostor pro řízení je na pravé straně. Druhý obrázek ukazuje okno s mapou a navigací vozidla po dojetí do stanoveného cíle. Na mapě je vidět jak samotné vozidlo, tak historie trasy kterou absolvovalo.



Obr. 5: Obrazovky ovládacího panelu

2.1 Architektura vzdáleného řídicího systému

Jednotka vzdáleného řídicího systému využívá stejnou hardwarovou architekturu jako řídicí jednotka vozidla (i.MX31 LiteKit s ARM architekturou). Jádrem je opět Freescale i.MX31 procesor. Vzdálený řídicí systém komunikuje s vozidlem přes WiFi modul OWSPA311 připojený na sériový port. Řídicí systém používá dotykový fólii umístěnou na barevném LCD displeji s rozlišením 640×480 bodů a 262 tisíci barvami. Je možné použít gamepad nebo joystick v případě manuálního módu řízení. Architektura je na obrázku 6.



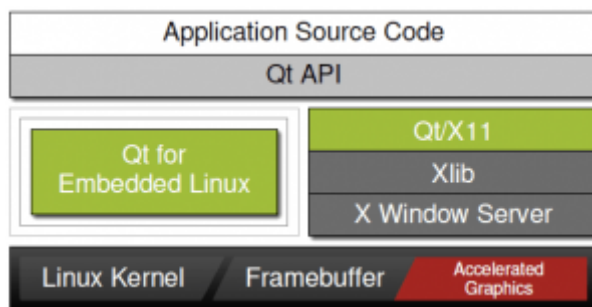
Obr. 6: Obrazovky ovládacího panelu

2.2 Softwarová architektura vzdáleného řízení

Je možné použít operační systém TimesysLinux jako v předchozím případě. Do jádra operačního systému musela být přidána podpora dotykové fólie, displeje a joysticku.

Grafická aplikace je vyvíjena ve Qt grafickém nástroji Qt Creator v programovacím jazyce C++ . V tomto nástroji je možné vytvořit grafický návrh aplikace (okna, tlačítka, texty). Pro vytvoření konečné aplikace je nezbytné vybrat externí kompilátor a přeložit aplikaci. Je možné také použít jiný způsob, například Qt Designer, který pouze vygeneruje grafiku a funkce. Zbytek aplikace je pak programován a přeložen v nějakém jiném IDE prostředí – TimeStorm v případě i.MX31.

Qt je rozšířenou grafickou knihovnou, která může být použita pro několik operačních systémů (Windows, X11, Linux). Knihovna je zjednodušena pro vestavěné (embedded) systémy. Šetří paměť, protože všechny potřebné funkce a knihovny má v sobě zahrnutý a zapisuje přímo do framebufferu.



Obr. 7: Architektura Qt embedded Linux

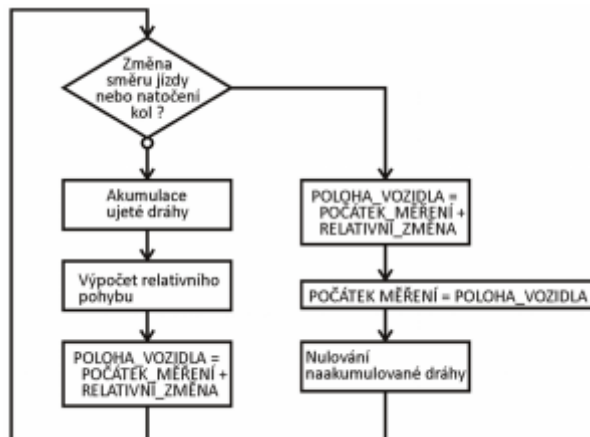
Aby mohla být grafická aplikace spuštěna, je nutné dodat grafické knihovny do souborového systému. Pro svůj provoz nevyžaduje spuštění externího grafického serveru, protože je integrovaný v knihovně. Použité procesy:

- vehicle_init - inicializace aplikace
- vehicle_guardian - hlídač spuštěných procesů (v případě zatumnutí, restartuje)
- vehicle_memory - proces sdílené paměti
- remote_gamepad - manuální řízení
- vehicle_wifi - komunikace se vzdáleným řídicím systémem
- panel - grafická aplikace ve knihovně Qt

Struktura procesů vzdáleného řídicího systému je podobná jako řídicího systému průzkumného vozidla. Navíc je zde proces remote_gamepad pro manuální řízení a panel, jako samotné grafická aplikace s uživatelským rozhraním.

3. Lokalizace pomocí odometrie

Jiná důležitá část projektu je založena na lokalizaci v budovách. Základní lokalizace je potřebná pro určení pozice uvnitř budovy. Proto je nutné načíst mapu detekované lokace do mobilního zařízení. WiFi signál není dostupný na všech místech. Na těchto místech je použita kombinace metod odometrie a jiných senzorů. Vozidlo jednoduše počítá trasu z inkrementálních senzorů na všech kolech a z úhlu natočení. Ve spolupráci se senzory měřící vzdálenost a mapou, je možné zlepšit nepřesnost této metody. Jednoduchý vývojový diagram ukazuje princip výpočtu odometrie.



Obr.8: Výpočet relativní změny polohy

4. Identifikace terénu a navigace

Analýza okolí a terénu je velmi důležitou funkcí při navigaci. Terén je monitorován převážně laserovým senzorem Sick LMS 100, který poskytuje dostatečně kvalitní data. Dále jsou kontrolována data z analogových dálkoměrů a ultrazvukových čidel. Vozidlo je navigováno po souřadném systému [x, y, natočení]. Data o aktuální pozici poskytuje modul polohy. Vozidlo má nahrané vektorové mapy a podle pozice z modulu polohy ví, kde se nachází.

Vozidlu je možné zadat pozici, kam má dojet. Řídící jednotka tuto informaci vyhodnotí a na pozici dojde. Nejprve dojde k návrhu trajektorie. Vektorové mapy obsahují informace o vhodných trasách, podobně jako GPS navigace v automobilech. Pomocí Dijkstrova algoritmu navrhne nejkratší cestu a po té se po ní vydá. Cestou vozidlo může narazit na překážky. Ty musí být překonány - objety. Průzkumné vozidlo je vždy naváděno po úsecích na bod a když je tento bod dosažen, je vozidlo naváděno na další.

Při detekci překážky na trase je volena objížďka. Vyhodnotí se prostor vpravo a vlevo a do sady bodů, na které je vozidlo naváděno je přidán další bod, který zajistí objetí překážky. Zároveň pokud je překážka velmi blízko naváděnému bodu, je tento bod přeskočen. Shrňme tedy navádění. Vozidlo se pohybuje po předem vypočítané trajektorii, která je složena z úseček resp. jejich koncových bodů. Zároveň cestou kontroluje překážky a překonává je. Navíc je na základě délky úseku a charakterem okolí vypočítávána bezpečná rychlost vozidla tak, aby nedošlo ke kolizi.



Video 1. Průzkumné vozidlo v pohybu

Literatura

1. Qt for Embedded Platforms [online]. 2010 [cit. 2010-04-13]. Qt for Embedded Platforms. Dostupné z WWW: <<http://doc.qt.nokia.com/4.6/qt-embedded.html>>
2. Kotzian, J., Srovnal, V.: Distributed embedded system for ultralight airplane monitoring. In sborníku, Anger, France: ICINCO, 2007, vol. 2007/1, p. 448-451, ISBN 978-972-88-5-82-5.
3. CiA [online]. 2010 [cit. 2010-03-17]. CAN in Automation. Available from WWW: <<http://www.can-cia.org/>>.
4. Zoom [online]. 2010 [cit. 2010-03-17]. Freescale Zoom i.MX31 LITEKIT. Available from WWW: <<http://www.logicpd.com/products/development-kits/freescale-zoom%E2%84%A2-imx31-litekit>>.
5. OWSPA311g: Electrical and mechanical datasheet. version 1.13. Sweden: [s.n.], 2008. 46s.
6. Timesys [online]. 2010 [cit. 2010-03-17]. LinuxLink. Available from WWW: <<https://linuxlink.timesys.com/3/Products>>.
7. Timesys [online]. 2010 [cit. 2010-03-17]. LinuxLink. Available from WWW: <<https://linuxlink.timesys.com/3/Products/TimeStorm>>.

Spoluautormi článku sú Tomáš LIPPA, Marian KURUC, Hynek PROKOP, Fakulta elektrotechniky a informatiky VŠB-TUO, Studentská tvůrčí a odborná činnost 2010, FAI UTB ve Zlíně
